IN THE CLAIMS:

Please amend the pending claim(s) as follows, substituting any amended claim(s) for the

corresponding pending claim(s):

1. (amended) For use in a data processor, a floating point unit comprising:

a plurality of floating point processing units capable of executing floating point

instructions that write operands to an external memory and capable of executing floating point

instructions that read operands from said external memory; and

an operand queue capable of storing a plurality of operands associated with one or

more operations being processed in said floating point unit, wherein said operand queue stores a first

operand written by a floating point write instruction executed by a first one of said plurality of

floating point processing units and wherein said operand queue supplies said first operand to a

floating point read instruction executed by a second one of said plurality of floating point processing

units when said first operand is committed or virtually committed if said floating point read

instruction requires said first operand.

1   2. (amended) The floating point unit as set forth in Claim 1 wherein said floating point unit

2   further comprises a store conversion unit capable of converting operands in said plurality of floating

3   point processing units from an internal format associated with said plurality of floating point

4   processing units to an external format associated with an external memory.

1   3. (unchanged) The floating point unit as set forth in Claim 2 wherein said operand queue

2   receives said first operand from said store conversion unit and transfers said first operand to said

3   external memory.

1   4. (amended) The floating point unit as set forth in Claim 1 wherein said floating point unit

2   further comprises a load conversion unit capable of converting incoming operands received from

3   said external memory from an external format associated with an external memory to an internal

4   format associated with said plurality of floating point processing units.

1   5. (unchanged) The floating point unit as set forth in Claim 4 wherein said operand queue

2   receives said incoming operands from said external memory and transfers said incoming operands

3   to said load conversion unit.

1   6. (unchanged) The floating point unit as set forth in Claim 5 wherein data in said external

2   memory is accessed in groups of N bytes and wherein said floating point unit further comprises at

3   least one aligner capable of receiving a first incoming operand that is misaligned with respect to a

4   boundary between a first N byte group and a second N byte group and aligning said first incoming

5   operand.

1   7. (unchanged) The floating point unit as set forth in Claim 6 wherein said operand queue

2   receives said aligned first incoming operand from said at least one aligner.

1   8. (unchanged) The floating point unit as set forth in Claim 7 wherein said at least one

2   aligner sets at least one bit in said operand queue to indicate that said aligned first incoming operand

3   is valid.

1   9. (amended) A data processor comprising:

2   at least one pipelined integer execution unit;

3   a data cache;

4   an instruction cache; and

5   a floating point unit comprising:

6   a plurality of floating point processing units capable of executing floating

7 point instructions that write operands to an external memory and capable of executing

8 floating point instructions that read operands from said external memory; and

9   an operand queue capable of storing a plurality of operands associated with

10 one or more operations being processed in said floating point unit, wherein said operand

11 queue stores a first operand written by a floating point write instruction executed by a first

12 one of said plurality of floating point processing units and wherein said operand queue

13 supplies said first operand to a floating point read instruction executed by a second one of

14 said plurality of floating point processing units when said first operand is committed or

15 virtually committed if said floating point read instruction requires said first operand.

1   10. (amended) The data processor as set forth in Claim 9 wherein said floating point unit

2   further comprises a store conversion unit capable of converting operands in said plurality of floating

3   point processing units from an internal format associated with said plurality of floating point

4   processing units to an external format associated with an external memory.

1   11. (unchanged) The data processor as set forth in Claim 10 wherein said operand queue

2   receives said first operand from said store conversion unit and transfers said first operand to said

3   external memory.

1   12. (amended) The data processor as set forth in Claim 9 wherein said floating point unit

2   further comprises a load conversion unit capable of converting incoming operands received from

3   said external memory from an external format associated with an external memory to an internal

4   format associated with said plurality of floating point processing units.

1   13. (unchanged) The data processor as set forth in Claim 12 wherein said operand queue

2   receives said incoming operands from said external memory and transfers said incoming operands

3   to said load conversion unit.

1   14. (unchanged) The data processor as set forth in Claim 13 wherein data in said external

2   memory is accessed in groups of N bytes and wherein said floating point unit further comprises at

3   least one aligner capable of receiving a first incoming operand that is misaligned with respect to a

4   boundary between a first N byte group and a second N byte group and aligning said first incoming

5   operand.

1   15. (unchanged) The data processor as set forth in Claim 14 wherein said operand queue

2   receives said aligned first incoming operand from said at least one aligner.

1   16. (unchanged) The data processor as set forth in Claim 15 wherein said at least one aligner

2   sets at least one bit in said operand queue to indicate that said aligned first incoming operand is valid.

17. (amended) For use in a floating point unit comprising a plurality of floating point

2   processing units capable of executing floating point instructions that write operands to an external

3   memory and capable of executing floating point instructions that read operands from the external

4   memory, a method of accessing the operands comprising the steps of:

5   storing in an operand queue a first operand written by a floating point write

6   instruction executed by a first one of the plurality of floating point processing units; and

7   supplying the first operand from the operand queue to a floating point read instruction

8   executed by a second one of the plurality of floating point processing units when the first operand

9   is committed or virtually committed if the floating point read instruction requires the first operand.

1   18. (amended) The method as set forth in Claim 17 wherein the floating point unit further

2   comprises a store conversion unit capable of converting operands in the plurality of floating point

3   processing units from an internal format associated with the plurality of floating point processing

4   units to an external format associated with an external memory.

1   19. (unchanged) The method as set forth in Claim 18 including the further steps of:

2   storing the first operand from the store conversion unit into the operand queue; and

3   transferring the first operand from the operand queue to the external memory.

C.ont
B1

ATD

1   20. (amended) The method as set forth in Claim 17 wherein the floating point unit further

2   comprises a load conversion unit capable of converting incoming operands received from an external

3   memory from an external format associated with the external memory to an internal format

4   associated with the plurality of floating point processing units.

1   21. (unchanged) The method as set forth in Claim 20 including the further steps of:

2   storing the incoming operands from the external memory in the operand queue; and

3   transferring the incoming operands from the operand queue to the load conversion

4   unit.